

Universidad Autónoma de Madrid

Escuela Politécnica Superior



Grado en Ingeniería Informática

TRABAJO DE FIN DE GRADO

**INTEGRACIÓN DE TECNOLOGÍAS DE ADQUISICIÓN Y
PROCESAMIENTO DE SEÑAL OCULAR PARA SU USO EN
APLICACIONES BIOMÉDICAS**

**María Godín Lorenzo
Tutor: Pablo Varona Martínez**

julio de 2018

INTEGRACIÓN DE TECNOLOGÍAS DE ADQUISICIÓN Y PROCESAMIENTO DE SEÑAL OCULAR PARA SU USO EN APLICACIONES BIOMÉDICAS

Autor: María Godín Lorenzo
Tutor: Pablo Varona Martínez

Grupo de Neurocomputación Biológica
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

julio de 2018

Agradecimientos

En primer lugar, dar gracias a mis padres, en especial a mi madre, por implicarse tanto y preocuparse porque todo vaya bien.

A Pablo, por darme la oportunidad de trabajar con él, por su paciencia y sus ánimos.

A mis compañeros y amigos de la universidad, Carli, Pelo, Guille, Edu y otros tantos que me han acompañado en esta etapa de mi vida de cerca.

A mis amigos de toda la vida, Fer, Carlota, gracias por el apoyo que me habéis dado, por estar ahí incondicionalmente, por quererme como me queréis.

A Sara y Quiris, por ser capaces de entenderme tan bien como lo han hecho y por los momentazos que hemos vivido, diversión, llantos y comprensión absoluta las unas con la otras.

A Carlos, por preocuparse de mi y de este proyecto desde casi el primer día hasta el último. Muchísimas gracias por ser tan amor.

A Álvaro, Barbas, Defessus. Gracias por aportarme la energía que solo tú me aportas, por escucharme y entenderme. Gracias por todo tu cariño durante todos estos años, sobretodo este último.

Gracias a todos los que han estado y se han quedado.

Y gracias a ti por leer esto.

Abstract

Abstract — The eye tracking is nowadays a versatile technology used in a lot of areas and sectors of the society, like for example psychology, marketing or medicine. Although there are several eye tracking data recording techniques, the main one consists in detecting a reflection of an infrared light on the user's cornea while performing an activity that requires eye movement.

This Final Degree Project has as a goal facilitating the use of eye tracking technologies in society, particularly in the context of detecting pathologies that have a high rate of patients not diagnosed.

In this Final Degree Project we have designed and developed an application to reach this objective. This application was developed in the programming language Python, and it is thought to can be used with a wide amount of eye tracking devices, in which we include since high range ones to low cost devices. That is the reason why in the development of this application we used the lowcost EyeTribe tracker device and the Tobii X2-30 tracker device.

For the development of the application we used the Python framework Kivy for the Graphical User Interface and the library named PyGaze for the communication between the eye tracking device and the application. We also used another libraries for other functionalities found in the application like could be audio recording or the generation of plots. The application has been validated in recording reading patterns prone to be used in the recommendation of diagnosis of diseases such as dyslexia or attention deficit.

Key words — eye tracking, dyslexia, attention deficit, pattern evaluation, pattern classification.

Resumen

Resumen — El seguimiento de la actividad ocular es actualmente una tecnología muy versátil y utilizada en múltiples campos y sectores, como pueden ser la psicología, el marketing o la medicina. Aunque existen diversas técnicas de registro de datos, la principal consiste en el seguimiento del reflejo de una luz infrarroja en la córnea del usuario mientras este realiza una actividad que requiera movimiento ocular.

Este Trabajo de Fin de Grado tiene como objetivo abordar el acercamiento de las tecnologías de seguimiento ocular a la sociedad, particularmente en el contexto de la recomendación de diagnóstico de enfermedades con una alta tasa de pacientes no diagnosticados.

Para conseguir este objetivo en este trabajo se ha desarrollado una aplicación en Python que puede utilizar una amplia gama de dispositivos de seguimiento ocular de distintas marcas y modelos para registrar la actividad ocular: desde dispositivos de alta gama a dispositivos de bajo coste, por ello, durante la implementación de la aplicación, hemos hecho uso de dos dispositivos de dos marcas diferentes. El dispositivo X2-30 de Tobii y el dispositivo *lowcost* de EyeTribe.

En el desarrollo de esta aplicación se ha utilizado el framework Kivy para el desarrollo de la interfaz gráfica y la librería Pygaze como intermediario entre la aplicación y los diferentes dispositivos de seguimiento ocular, además de otras librerías usadas para la grabación de audio y generación de gráficos. La aplicación se ha validado en el registro de patrones de lectura susceptibles de ser utilizados para la recomendación de diagnóstico de enfermedades como la dislexia y el déficit de atención.

Palabras clave — seguimiento ocular, dislexia, déficit de atención, evaluación de patrones de lectura, clasificación de patrones.

Índice general

1. Introducción	1
1.1. Motivación y objetivos	1
1.2. Alcance	2
1.3. Estructura del documento	2
2. Estado del Arte	5
2.1. Origen y Trayectoria del Seguimiento ocular	5
2.2. Seguimiento ocular	6
2.3. Dispositivos para seguimiento de vista	7
2.4. Dislexia	10
2.5. Uso de dispositivos de seguimiento ocular con fines médicos	11
2.6. Futuro del eye tracking	12
3. Diseño	13
3.1. Dispositivos	13
3.1.1. Colocación de los dispositivos	14
3.1.2. Calibración	15
3.1.3. Datos que proporciona	15
3.2. Usuarios	16
3.3. Aplicación	16
3.3.1. Distribución	17
3.4. Ficheros de lectura	18
3.5. Grabación de audio	18
3.6. Calibración	18
3.7. Resultados	18
4. Desarrollo	19
4.1. Herramientas usadas	19
4.1.1. PyGaze	19
4.1.2. Kivy	20
4.1.3. PyGame	20
4.1.4. PyAudio	20
4.1.5. Matplotlib	21
4.2. Aplicación	21

4.2.1. Registro de usuario	21
4.2.2. Personalización de Configuración	21
4.2.3. Calibración y prueba	21
4.2.4. Generación de resultados	22
4.3. Interfaz de la aplicación	22
4.3.1. Interfaz final	22
5. Resultados	27
5.1. Objetivos de la evaluación	27
5.2. Pruebas	27
5.3. Resultados	28
5.3.1. Voluntario 1	28
5.3.2. Voluntario 2	29
5.3.3. Voluntario 3	31
5.3.4. Voluntario 4	33
5.3.5. Voluntario 5	34
5.4. Análisis de los resultados	36
6. Conclusiones	37
6.1. Conclusiones	37
6.2. Trabajo futuro	37
Bibliografía	39
Apéndices	41
A. Extensión de diseño	43
A.1. Requisitos	43
A.1.1. Requisitos funcionales	43
A.1.2. Requisitos no funcionales	44
A.2. Diagrama de clases	44
A.3. Código obtenido	44
B. Guía de Instalación	47
B.1. Interfaz gráfica	47
B.2. Eyetracking	47
B.3. Audio	48
B.4. Generación de gráficas	49
C. Información extendida de los eye trackers usados	51
C.1. Comparación de eye trackers	51
C.2. SDK	51
C.2.1. EyeTribe	52
C.2.2. Tobii	52

Índice de tablas

3.1. Datos obtenidos a partir de los dispositivos de seguimiento ocular	16
C.1. Tabla de comparación de los dispositivos utilizados durante la prueba	51

Índice de figuras

2.1. Eye tracker de Buswell	6
2.2. Actividad ocular de un usuario	7
2.3. Ejemplo de eye tracker intrusivo	8
2.4. Técnicas de seguimiento ocular	9
2.5. Posición de la luz infrarroja relativa a la posición de la pupila	10
2.6. Comparación del movimiento ocular entre dos lectores	11
3.1. Dispositivo de seguimiento ocular The EyeTribe.	13
3.2. Dispositivo de seguimiento ocular Tobii X2-30.	14
3.3. Colocación de dispositivo de eyetracking no integrado.	14
3.4. Ejemplo de calibración realizado con nuestra aplicación.	15
3.5. Crecimiento de los lenguajes de programación	17
4.1. Ventana de inicio	23
4.2. Registro de usuario	23
4.3. Ventana de información adicional	24
4.4. Ventana de configuración	25
4.5. Calibración del eye tracker EyeTribe	25
4.6. Calibración del eye tracker Tobii	26
5.1. Tamaño de pupila	28
5.2. Coordenadas de ambos ojos	28
5.3. Coordenada de ojo izquierdo	29
5.4. Coordenada de ojo derecho	29
5.5. Tamaño de pupila	30
5.6. Coordenadas de ambos ojos	30
5.7. Coordenada de ojo izquierdo	30
5.8. Coordenada de ojo derecho	31
5.9. Tamaño de pupila	31
5.10. Coordenadas de ambos ojos	32
5.11. Coordenada de ojo izquierdo	32
5.12. Coordenada de ojo derecho	32
5.13. Tamaño de pupila	33
5.14. Coordenadas de ambos ojos	33
5.15. Coordenada de ojo izquierdo	34

5.16. Coordenada de ojo derecho	34
5.17. Tamaño de pupila	35
5.18. Coordenadas de ambos ojos	35
5.19. Coordenada de ojo izquierdo	35
5.20. Coordenada de ojo derecho	36

1

Introducción

1.1. Motivación y objetivos

Los avances de la tecnología y su integración en la sociedad ayudan a la mejora de la calidad de vida de las personas y permiten facilitar tareas, mejorar la adquisición y gestión de datos y su correspondiente análisis, y proporcionar soluciones a problemas con fuerte impacto socio-económico. Esta tendencia, a su vez, provoca que la tecnología cada vez sea más accesible y de menor coste.

La tecnología de seguimiento ocular se encuentra actualmente en auge, siendo usada en una gran cantidad de áreas y disciplinas [1, 2, 3]. Los dispositivos de seguimiento ocular, o eye trackers, son los encargados de recolectar los datos e información obtenidos de la actividad ocular. Estos dispositivos pueden variar tanto en las técnicas usadas para la obtención de los datos como en el precio, habiendo eye trackers de un coste muy reducido, como el modelo de EyeTribe, cuyo precio es de 100\$, o de un coste mayor, por ejemplo, el modelo de Tobii Tx-300 que alcanza los 45.000€

La lectura es una habilidad imprescindible para desarrollarse correctamente y de manera independiente en la sociedad. Cuando no se desarrollan correctamente estas habilidades, ya sea por trastornos que provocan trabas en la atención o en la comprensión de los textos o por falta de formación elemental, pueden surgir varios problemas en el desarrollo de la persona.

La motivación de este Trabajo de Fin de Grado es favorecer la difusión de las tecnologías de seguimiento ocular, haciendo hincapié en los dispositivos y sistemas *lowcost* con el fin de facilitar la detección de trastornos que tengan una clara manifestación en los movimientos oculares.

Este trabajo parte de un desarrollo previo implementado por Víctor Díaz Herraes en su Trabajo de Fin de Grado, en el cual desarrolló una aplicación cuyo objetivo era la comparación de distintos eye trackers. La gran diferencia con este proyecto es tanto la motivación del mismo como su implementación, al estar su aplicación desarrollada en el lenguaje de programación C#, el cual hemos decidido sustituir por Python, al considerarlo un lenguaje mucho más versátil y que otorgará más flexibilidad al futuro de nuestra aplicación.

Los objetivos de este Trabajo de Fin de Grado son los siguientes:

- El diseño de una aplicación capaz de realizar seguimiento ocular para su uso en dispositivos de seguimiento ocular de distintos fabricantes.
- Diseñar las herramientas de aplicación para su uso en el registro de patrones de lectura.
- Implementar la aplicación y las herramientas asociadas en Python para favorecer su uso en una amplia gama de aplicaciones y sistemas.
- Validar las herramientas desarrolladas en dos eye trackers: uno de la marca EyeTribe y otro de Tobii.

1.2. Alcance

El alcance de este Proyecto es la llegada a distintas instituciones como escuelas, colegios u hospitales, donde se pueda aprovechar la aplicación para facilitar la detección de patrones oculares anómalos.

El objetivo es que, haciendo uso de un eye tracker de bajo coste y nuestra aplicación, lo que implicaría la compra de un ordenador y un eye tracker *lowcost* (entre 100€ y 1000€) pueda usarse para detectar con más facilidad patologías como la dislexia y el déficit de atención a partir del registro de patrones de lectura.

1.3. Estructura del documento

El documento se organizará en los siguientes capítulos:

Estado del arte. En este capítulo se realizará un breve recorrido histórico con respecto al seguimiento ocular, después se realizará una explicación sobre el seguimiento ocular, los dispositivos que realizan este seguimiento y las distintas técnicas que utilizan. También se hablará sobre la dislexia, al ser una enfermedad que afecta a la atención y es detectable a partir de patrones oculares. Finalmente hablaremos sobre la evolución actual que están teniendo las tecnologías y dispositivos de seguimiento ocular.

Diseño. En este capítulo se explicará el diseño realizado y distintas razones por las que se ha decidido así.

Desarrollo. En este apartado se mostrará el funcionamiento de la aplicación y explicando las distintas opciones que ofrece.

Pruebas y resultados. Se comentarán los resultados obtenidos al probar la aplicación en distintos voluntarios.

Conclusiones. Se realizará una conclusión con respecto al trabajo realizado y se desarrollará una idea con respecto al trabajo futuro.

2

Estado del Arte

2.1. Origen y Trayectoria del Seguimiento ocular

La atención visual del ser humano ha sido materia de estudio desde hace siglos, basándose inicialmente en observaciones del ojo en la mayoría de los casos y en introspección en algunos de ellos [2].

El seguimiento ocular comenzó su trayectoria en la historia en el siglo XIX, realizándose inicialmente a través de observaciones del ojo. En el año 1879, el oftalmólogo francés Louis Émile Javal, observó cómo el movimiento de los ojos de una persona no seguía trayectorias fluidas con la lectura de textos [4].

En 1908, Edmund Huey construyó el que se puede considerar el primer aparato de seguimiento ocular durante la lectura de un individuo. Este eye tracker, a diferencia de los actuales, era muy intrusivo con el usuario al se le estaba analizando la vista.

Durante esta época se descubrieron las características básicas sobre el movimiento ocular, como la supresión sacádica, la latencia de la sacada y el tamaño de la ventana de percepción[4].

Guy Thomas Buswell construyó el primer eye tracker no intrusivo de la historia, consistente en reflejar haces de luz en los ojos del lector, siendo este proceso grabado en una cinta de vídeo [5].

Entre las décadas de los 70 y los 80, marcados por la mejora de las técnicas de grabación que permitían conseguir resultados de una forma más rápida y más precisa [6].

Muchas teorías psicológicas comenzaron a considerar un factor a tener en cuenta los

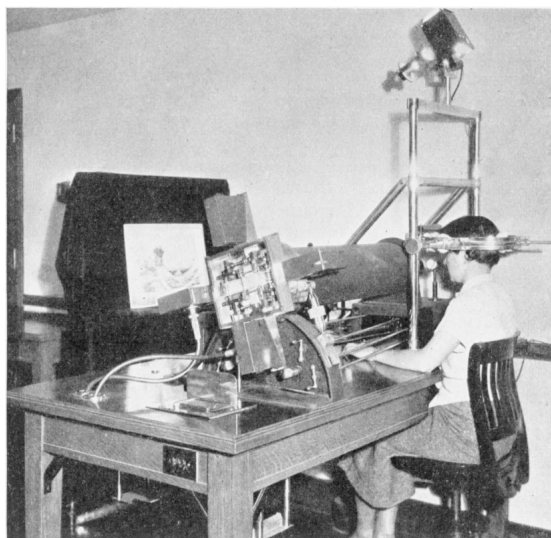


Figura 2.1: Eye tracker de Buswell

<http://eyeseer-research.com/blog/eye-tracking-history/>

patrones de vista para el análisis los procesos cognitivos. Además, en los años 80, se comenzó a usar en el área del marketing, para analizar qué elementos se observan en un anuncio publicitario [7].

2.2. Seguimiento ocular

El movimiento ocular se puede dividir en diferentes componentes:

Fijaciones. Paradas de entre 200 y 300 ms. Es el donde más información adquiere el sujeto. Todos los eye trackers proporcionan datos de la parada, consistentes en tiempo de duración y una coordenada.[8]

Sacadas. Movimiento rápido de los ojos realizado entre las fijaciones voluntaria o involuntariamente. Pueden durar de 10 a 100 milisegundos. En esta acción, los procesos cognitivos y la codificación de información son muy limitados.[8] Pueden llegar hasta 500° [9].

Regresión. Acción de volver a la parte anterior de un texto para revisar información [10].

Ventana de percepción. Se trata del área observada durante una fijación. En el caso de estar observando un texto, sería la cantidad de palabras o letras observadas durante una fijación. En el caso de los lectores con alfabetos leídos de izquierda a derecha, la ventana de percepción se suele extender entre 3 y 4 espacios a la izquierda desde el punto de fijación y 14 y 15 a la derecha.[11]

Áreas de interés. Áreas de estimulación. Se dice que un área de interés ha sido visitada cuando se ha realizado al menos una fijación en ella.

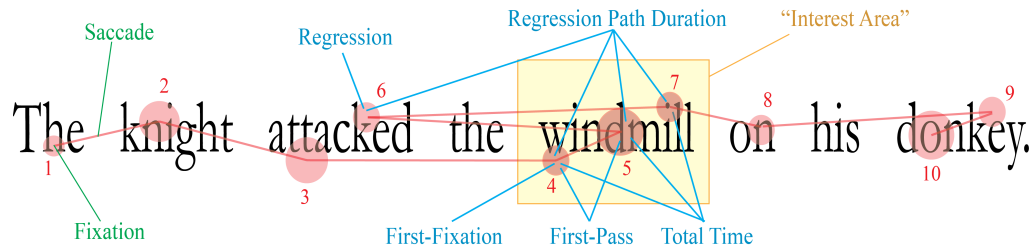


Figura 2.2: Actividad ocular de un usuario
<http://www.icge.co.uk/languageciencesblog/?p=1216>

Las técnicas de seguimiento ocular permiten observar tanto las sacadas y fijaciones como las áreas de interés.

Actualmente, el seguimiento ocular se utiliza para multitud de disciplinas, entre las que destacamos el marketing, estudios en el área cognitiva y psicología, diseño industrial y ciencias de la computación.

La relación entre el movimiento ocular y la comprensión se basa en dos suposiciones

- En primer lugar, nos basamos en que, en cuanto se observa un estímulo, se trata de interpretar inmediatamente.
- En segundo lugar, que no se deja de prestar atención a un estímulo (ser objeto de fijaciones) hasta que este se comprende.

2.3. Dispositivos para seguimiento de vista

Los dispositivos para el seguimiento de la vista se conocen como eye trackers.

Los eye trackers han sido diseñados para monitorizar la atención visual con la recopilación de movimientos oculares cuando se observan diferentes estímulos.

Tras todos los estudios realizados durante el siglo pasado, se ha conseguido aportar gran diversidad de metodologías con las que conseguir información[11]. Existen dos técnicas de monitorización de movimiento, una que mide la posición del ojo en relación a la cabeza, y otra que mide la orientación del ojo en el espacio, o el punto de visión.

Dentro de las categorías de eye trackers, podemos encontrar dos tipos:

Intrusivos. Constan de una cinta con tres cámaras que se tiene que poner el participante durante la prueba. Dos de las cámaras realizan una grabación de los movimientos

oculares haciendo uso de haces de luz infrarroja reflejada en la pupila mientras que la otra se encarga de hacer un seguimiento de la cabeza, pudiendo ser optativa. Las desventajas de este tipo de dispositivos de seguimiento de vista son, en primer lugar, que un movimiento de cabeza puede provocar una descoordinación entre la cámara y el ojo y que puede provocar cierta incomodidad del participante al tener que llevar puesta la cinta encima. [8]

No intrusivos. Con respecto a esta categoría, podemos hacer una división temporal entre los más antiguos y los más modernos. Los más antiguos hacían uso de haces de luz que se reflejaban en los ojos del participante y capturados en su retorno. Los más actuales, usan para su funcionamiento un ordenador, dos cámaras y un detector infrarrojos. El ordenador detecta y graba el movimiento de los ojos haciendo referencia de las diferentes partes de la cara. Además hace uso de los rayos infrarrojos reflejados en la córnea para diferenciar los movimientos de cabeza de los de pupila.[8]



Figura 2.3: Ejemplo de eye tracker intrusivo
<http://www.uta.fi/sis/tauchi/virg/laboratory.html>

Los dispositivos de seguimiento ocular hacen uso de diferentes técnicas para conseguir realizar correctamente su función. Las técnicas más usadas son las descritas a continuación:

Electrooculografía. Consistente en colocar electrodos alrededor del ojo y registrar las diferencias de potencial eléctrico de la piel. Esta metodología fue la más utilizada durante los años 70. La desventaja de esta técnica es que la medición del movimiento ocular es relativa a la posición de la cabeza, no siendo por tanto una técnica válida para la medición del punto de visión [2].

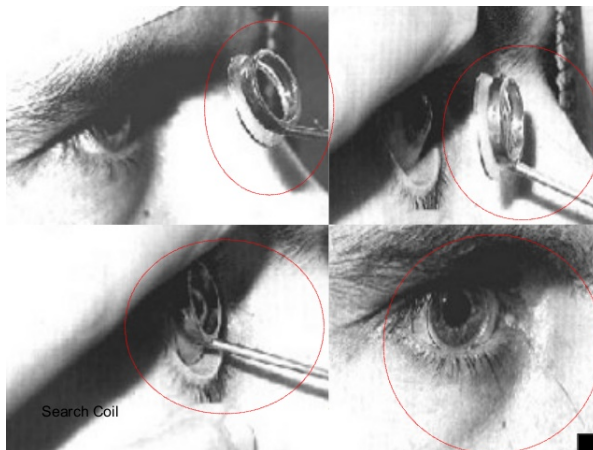
Lentillas esclerales. Consiste en utilizar unas lentes de contacto con una serie de componentes que permiten realizar un seguimiento del movimiento del ojo, como fósforos reflectantes o alambres. Estas lentes de contacto son más grandes de lo

normal, cubriendo tanto la cornea como la esclerótica. Permite una medición de los movimientos oculares muy precisa. La parte negativa del uso de esta técnica es, por un lado que poner las lentillas al sujeto es una tarea muy delicada que precisa práctica y cuidado, además de que resultan muy incómodas, que la posición ocular es relativa a la cabeza y no es el mejor método para la medición del punto de visión [2].



(a) Ejemplo de electrooculografía

<https://www.slideshare.net/manu8722487734/eie426-eye-trackingv2-1>



(b) Colocación de lentillas esclerales

Figura 2.4: Técnicas de seguimiento ocular

<https://www.slideshare.net/manu8722487734/eie426-eye-trackingv2-1>

Fotoculograma y videoculograma. En esta categoría se contempla una gran variedad de técnicas de grabación de las distintas características distinguibles del ojo durante la rotación y la traslación (forma de la pupila, el reflejo de la luz infrarrojos en la córnea, etc.). Estas técnicas no realizan una medición del punto de visión [2].

Técnica basada en vídeo combinada con reflejo en la córnea y la pupila. Como hemos observado anteriormente, las técnicas anteriores, aunque aporten muchos datos, ninguna realiza la medición del punto de visión. Para obtener esta medición la cabeza debe de estar fija, y los movimientos oculares coincidir o realizar múltiples medidas de diferentes características oculares para permitir diferenciar el movimien-

to ocular del de la cabeza. Dos de estas características son la reflexión en la córnea y el centro de la pupila [2].



Figura 2.5: Posición de la luz infrarroja relativa a la posición de la pupila
<https://usabilitygeek.com/what-is-eye-tracking-when-to-use-it/>

Estas técnicas utilizan cámaras de un coste relativamente reducido y un hardware de procesamiento de imagen capaz de medir el punto de mira en tiempo real.

2.4. Dislexia

La dislexia es una discapacidad neurológica que afecta a las capacidades de escritura y lectura de un individuo, al provocar una dificultad en la precisión de reconocimiento de palabras y una pobre capacidad de deletreo y decodificación [12]. Se suele asociar con una memoria frágil y con dificultades con el procesamiento visual y auditivo[9]. Aproximadamente un 10 % de la población padece dislexia [12].

Un niño que padece dislexia, muestra dificultades con respecto al aprendizaje de habilidades lectoras aún mostrando un perfil neurológico, cognitivo y emocional aparentemente normales [13].

Se han realizado varios estudios para obtener información sobre el comportamiento visual en niños disléxicos, que consistían en realizar pruebas sobre habilidades oculomotoras muy básicas, en personas que sufren esta discapacidad. Estos estudios demuestran las diferencias entre los patrones de lectura de una persona sana y otra disléxica [12].

Uno de estos estudios muestra como un lector disléxico realiza con mayor latencia tanto las sacadas como la vergencia. Otros autores también han observado como los niños disléxicos cuentan con peor coordinación binocular en las sacadas y en las fijaciones en la observación de cuadros y pinturas.

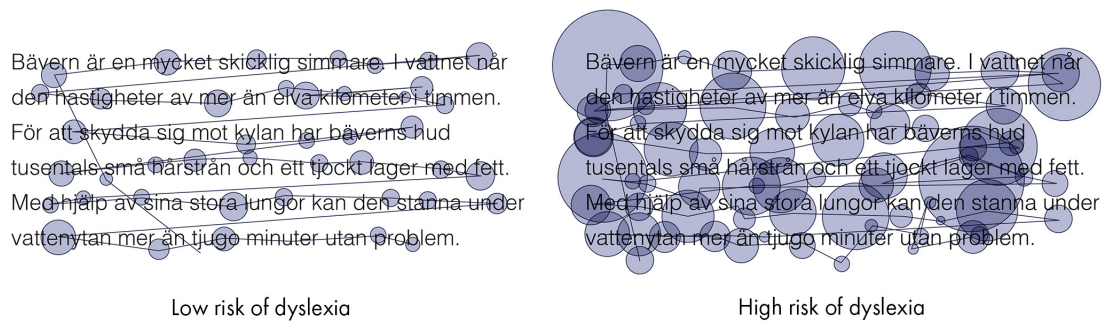


Figura 2.6: Comparación del movimiento ocular entre dos lectores
<https://www.accelify.com/accelify-blog/2017/11/28/ai-eye-tracking-soon-help-schools-screen-dyslexia/>

Además, la dislexia suele estar asociada a otras discapacidades de aprendizaje, del lenguaje y déficit de atención. [11]. No se conoce su causa concreta, pero la teoría más generalizada es que sea provocada por algún tipo de déficit en el procesamiento a nivel fonológico [11].

Las metodologías tradicionales para medir los grados de dislexia en niños suele realizarse con la manipulación de ritmos, sílabas y fonemas [9].

Un estudio típico para detectar la presencia de un déficit cognitivo subyacente es mostrando una cadena de caracteres y haciéndole elegir a la persona a la que se le están enseñando las mismas si considera que esa cadena es una palabra o no pulsando un botón.[14]

Estudios tanto psicológicos como visuales indican que el movimiento ocular de una persona disléxica varía con respecto al de una persona que no lo es en cualquier contexto, no solo en la lectura.

2.5. Uso de dispositivos de seguimiento ocular con fines médicos

Varios estudios demuestran que el movimiento ocular está relacionado con los procesos cognitivos mientras se está realizando la lectura de un texto. [11].

Las técnicas de seguimiento ocular pueden ser muy positivas en este campo, al aportar datos cuantitativos y objetivos de los procesos visuales y de la atención del paciente. Además, su uso es casi transparente para el paciente al ser tan discreta su forma de uso [15].

Actualmente, se están utilizando los dispositivos de seguimiento ocular para la detección de patologías y desordenes mentales. Tal y como hemos visto en el apartado

anterior, que se podían observar diferencias entre los patrones de lectura de personas disléxicas y no disléxicas.

Además de para las enfermedades que afectan a la atención, se utiliza el eyetracking para estudios que cubren distintas neuropatologías o desordenes psicologicos en adolescentes y niños. E incluso se utiliza para la detección temprana de enfermedades de carácter neurodegenerativo, como puede ser el Alzheimer, al contar con una sensibilidad de datos como para analizar las anormalidades oculomotoras de estas enfermedades [16].

Por tanto, el uso de estos dispositivos en el área de recomendación de diagnóstico resulta provechosa y productiva, al haber conseguido un mayor conocimiento de la relación entre la visión y la atención gracias a la investigación a nivel medico realizada con los eye trackers [11].

2.6. Futuro del eye tracking

Como se ha comentado anteriormente, los eye trackers tienen aplicaciones en diferentes áreas, aunque su uso es limitado por muchos factores, como su alto coste, un hardware invasivo o muy personalizado o falta de precisión bajo las condiciones del mundo real.

Una forma de acercamiento de las tecnologías de eye tracking en la sociedad podrían ser los teléfonos móviles. Se estima que en 2019, más de un tercio de la población contará con un smartphone.

Un ejemplo de acercamiento de estas tecnologías a la sociedad sería la generación de GazeCapture, un conjunto de datos con 1500 muestras de seguimiento ocular obtenido con distintos fondos y variaciones. Estos datos se utilizarán para entrenar una red neuronal convolucional que realizará una predicción de donde se posará la mirada, consiguiendo una predicción con un error de 1,04 cm en móviles y 1,69 en tablets. [7]

En este Trabajo de Fin de Grado, haremos uso de la tecnología descrita para el favorecimiento de la detección de patrones oculares anómalos que puedan ser signo de cualquier trastorno o enfermedad que tenga alguna representación en el movimiento ocular.

3

Diseño

En este apartado se especificará el diseño decidido para la aplicación. En el anexo A se mostrarán los requisitos de la aplicación el diagrama de clases final.

3.1. Dispositivos

Para la realización de esta aplicación se ha hecho uso de dos eye trackers. El dispositivo de EyeTribe y el modelo x2-30 de Tobii.

Los motivos de uso de estos dos dispositivos han sido, además de su disponibilidad (al encontrarse los dos en el laboratorio del Grupo de Neurocomputación Biológica) en caso del primero de ellos, su precio de venta (99 \$), tan competitivo que puede permitir su acceso a un conjunto de personas muy amplio y que podría permitir una generalización en el uso de los dispositivos de seguimiento ocular.



Figura 3.1: Dispositivo de seguimiento ocular The EyeTribe.

En el caso del segundo dispositivo, aunque cuenta con un precio muy superior (20000\$), se ha escogido debido a que su fabricante es muy popular y conocido en su campo, siendo muy posible que en un futuro sus dispositivos, desde un punto de vista económico, sean más cercanos a la sociedad.



Figura 3.2: Dispositivo de seguimiento ocular Tobii X2-30.

Para más información con respecto a estos eye trackers en el anexo C se adjunta una tabla comparando las distintas características entre ellos.

3.1.1. Colocación de los dispositivos

Ambos eye trackers son no integrados.

Para un correcto funcionamiento de un dispositivo no integrado, este se debe de colocar en la parte inferior de la pantalla, lo más centrado posible. Se conectará al ordenador vía USB, en el caso del dispositivo de Tobii, tiene que conectarse obligatoriamente a un USB 2.0 mientras que en el caso del dispositivo EyeTribe, neccesariamente debe ser un USB 3.0.

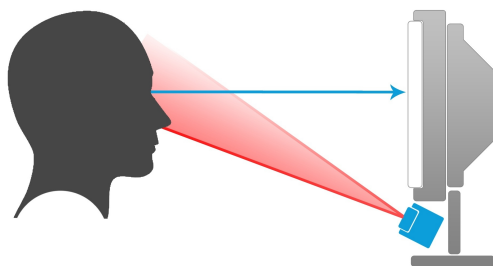


Figura 3.3: Colocación de dispositivo de eyetracking no integrado.

<http://recordency.gq/archive/Eye-tracking-mouse-freeware-downloads.html>

3.1.2. Calibración

El proceso de calibración se encarga de realizar una captura inicial del movimiento ocular, encargándose de dimensionarlo. Consistirá en la presentación de varios estímulos que el usuario debe mirar en distintos puntos de la pantalla y que el software del dispositivo utilizará como referencia, al conocer el tamaño de la pantalla.

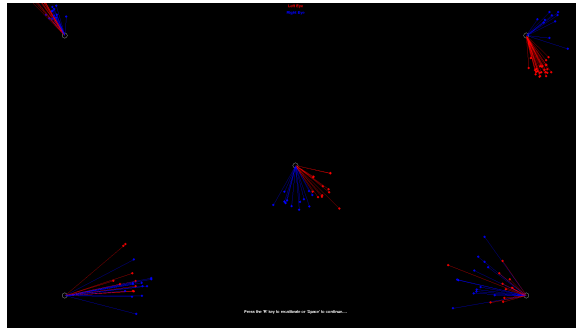


Figura 3.4: Ejemplo de calibración realizado con nuestra aplicación.

A nivel general, ambos dispositivos cuentan con programas para la calibración facilitados por los fabricantes. Para nuestra aplicación, haremos uso de la librería PyGaze, desarrollada para el manejo de dispositivos de seguimiento ocular, para realizar esta operación. Esta librería utilizará de base para la calibración las SDK facilitadas por los fabricantes. En el anexo C hemos facilitado una guía para obtener más información sobre su instalación.

3.1.3. Datos que proporciona

Nuestra aplicación genera dos ficheros con información sobre las muestras recogidas en una sola prueba. El primer fichero es un fichero generado por la SDK del eye tracker. El segundo, lo genera la librería PyGaze.

Cada uno de ellos tiene ventajas y desventajas en su uso. En el caso de los ficheros generados por las distintas SDK, la parte positiva es que permiten el acceso a muchos más datos que en el caso del fichero generado por la librería. La parte negativa es la heterogeneidad entre los distintos ficheros, al variar tanto los atributos mostrados como las unidades.

EyeTribe	TimeStamp Coordenada de pupila	Fijacion Datos para ojo izquierdo	Estado -	Coordenada Datos para ojo derecho	Coordenada Suavizada -	Tamaño de pupila -
Tobii	TimeStamp Coordenada de pupila	- Datos para ojo izquierdo	- Validación para coordenada del ojo izquierdo	Coordenada Datos para ojo derecho	- Validación para coordenada del ojo derecho	Tamaño de pupila -

Tabla 3.1: Datos obtenidos a partir de los dispositivos de seguimiento ocular

Nuestra aplicación en principio hará uso de las coordenadas, tanto de ambos ojos como solo del ojo izquierdo y del ojo derecho, y el tamaño de pupila.

Las coordenadas tanto de los dos ojos como solo del izquierdo y solo del derecho comparten la misma unidad en ambos ficheros, píxeles.

Las unidades del tamaño de pupila de ambos ojos en cambio varían. El dispositivo EyeTribe hace uso de píxeles para medir su tamaño, en cambio Tobii utiliza milímetros para su medición.

Para el desarrollo de nuestra aplicación hemos decidido usar los ficheros generados por las SDK al contener más información. Aunque creemos que provoca una mayor complicación con respecto al desarrollo de la aplicación, hemos considerado que esta mayor cantidad de datos puede resultar beneficiosa ante el diagnóstico.

3.2. Usuarios

La aplicación está orientada a un público muy genérico, con un conocimiento tecnológico no muy avanzado.

Los requisitos son tener un conocimiento mínimo de la prueba que esta realizando y ser capaz de realizar una lectura literal del texto.

3.3. Aplicación

La aplicación ha sido desarrollada íntegramente en el lenguaje de programación Python. Hemos escogido este lenguaje debido a su gran popularidad y uso en multitud de campos, lo que puede permitir un mejor desarrollo y evolución de esta aplicación en el futuro.

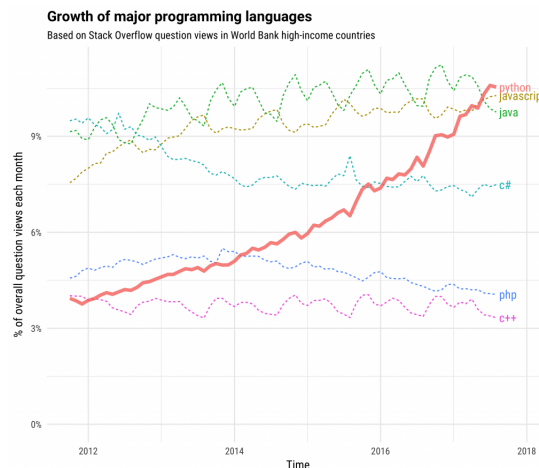


Figura 3.5: Crecimiento de los lenguajes de programación
<https://stackoverflow.blog/2017/09/06/incredible-growth-python/>

Además, se ha decidido trabajar sobre la distribución WinPython. Este entorno cuenta con múltiples ventajas que favorecen su uso. En primer lugar, una instalación sencilla a partir de un ejecutable. Cuenta con muchas librerías que usamos en este proyecto ya instaladas y configuradas. Es portable, se puede comprimir y volver a usar sin ningún problema en otro ordenador.

3.3.1. Distribución

Hemos dividido la lógica implementada en distintas carpetas y ficheros según las tareas a realizar y las áreas en las que trabaja.

El código de la aplicación se divide de la siguiente manera:

```

Aplicacion
├── application.py
├── graphic_engine ..... Implementación de la interfaz gráfica
│   ├── __init__.py
│   ├── layouts.py
│   └── backgrounds
├── eye_tracker ..... Manejo del eye tracking
│   ├── __init__.py
│   └── eyetrackerhandler.py
├── data_manipulator ..... Tratamiento de los datos extraídos
│   ├── __init__.py
│   └── datamanipulator.py
├── user ..... Configuración de nombres de ficheros
│   ├── __init__.py
│   └── user.py

```

3.4. Ficheros de lectura

Los ficheros utilizados para este proyecto son imágenes de formato PNG que contienen un texto con una alineación de 1,5mm. Además, esta imagen debe de tener el tamaño de la pantalla, en el caso del ordenador utilizado para realizar las pruebas 1900x1080.

Las imágenes utilizadas para la realización de las pruebas las obtenemos a partir del proyecto realizado por Víctor Díaz. El cual consistía en el desarrollo de una aplicación en C# para el análisis y la comparación entre dispositivos de seguimiento ocular.

3.5. Grabación de audio

La grabación de audio es un complemento a la captación de movimientos oculares que aporta mayor situación, contexto e información de la prueba.

Se realizará por ello una implementación de grabación de audio en la aplicación para obtener estos datos.

3.6. Calibración

La calibración se realizará a través de la aplicación, apoyándose en las SDK de cada uno de los eye trackers de forma interna.

3.7. Resultados

Los resultados obtenidos se extraerán de los ficheros generados. Tras realizar la prueba se mostrarán sus datos en varias gráficas al usuario.

- Gráfica de coordenada. Esta gráfica mostrará la trayectoria del movimiento de los dos ojos.
- Gráfica de coordenadas para el ojo izquierdo y el derecho respectivamente. Mostrará la trayectoria del movimiento ocular de cada uno de los ojos en dos gráficas distintas.
- Tamaño de pupila. Se generará una gráfica de la variación del tamaño de la pupila de ambos ojos a lo largo de la prueba.

4

Desarrollo

Este proyecto consta de una aplicación que mostrará gráficamente los datos recogidos por un dispositivo de eyetracking.

En este apartado comentaremos de forma detallada el funcionamiento de la aplicación.

4.1. Herramientas usadas

Para el desarrollo de esta aplicación hemos utilizado tanto herramientas externas a Python como librerías internas.

4.1.1. PyGaze

Para el manejo de los eye trackers hemos hecho uso de la librería de Python PyGaze [17].

Esta librería nos permite poder trabajar con múltiples dispositivos de seguimiento ocular de una forma genérica. Se han ido añadiendo varios dispositivos de distintas marcas desde su implementación inicial. Actualmente funciona con dispositivos Tobii, pudiendo usar la nueva SDK implementada en 2017 y la anterior a esta, EyeTribe, SMI, GazePoint y EyeLink. Para más información sobre esta librería y todas sus funcionalidades podemos acceder a su web <http://www.pygaze.org/>

Tobii

Bajo esta librería, utilizaremos la SDK de Tobii, que podemos encontrar en <https://www.tobiipro.com/product-listing/tobii-pro-sdk>. Esta SDK se puede utilizar para cualquier dispositivo Tobii.

EyeTrybe

La SDK de EyeTribe la podemos obtener en el siguiente enlace: <https://github.com/EyeTribe/sdk-installers/releases>.

4.1.2. Kivy

La interfaz gráfica se ha desarrollado con el framework Kivy. Este framework permite una implementación de aplicación "user friendly" multiplataforma con propiedades multitáctiles, lo que mantienen abierto un posible desarrollo futuro en smartphones o tablets.

Cada pantalla cuenta con multitud de configuraciones posibles.

Una peculiaridad de este framework es que permite al programador hacer uso de ficheros de extensión kv para facilitar la implementación de la interfaz. Permite crear un árbol de widgets de forma declarativa.

Toda la información para su instalación y configuración la podemos encontrar en su página web <https://kivy.org/#home>

4.1.3. PyGame

Con respecto a la parte de calibración y la prueba, se hará uso de la librería PyGame, gestionada a partir de la librería PyGaze.

Esta librería permite la creación de juegos e interfaces en dos dimensiones. En principio esta planteada para la realización de videojuegos, aunque es totalmente funcional para otras implementaciones como es el caso de esta aplicación.

Toda la información con respecto a esta librería la podemos encontrar en la pagina web <https://www.pygame.org>

4.1.4. PyAudio

Esta librería realiza grabaciones y reproducciones de audio. Hacemos uso de esta librería para la grabación de audio durante la prueba.

La información referente a esta librería la podemos encontrar en <https://people.csail.mit.edu/hubert/pyaudio/>

4.1.5. Matplotlib

Se trata de una librería de Python que genera gráficas en dos dimensiones. En nuestra aplicación la utilizamos para generar las gráficas con los datos extraídos en las pruebas.

Se puede obtener más información a través de la web <https://matplotlib.org/>

4.2. Aplicación

Hemos desarrollado la lógica interna de las acciones del usuario, el registro de datos personales, personalización de la configuración, calibración del eye tracker, la prueba y finalmente la generación de datos.

4.2.1. Registro de usuario

El usuario debe rellenar una serie de campos con datos personales (nombre, apellidos y edad). Para su implementación usamos la clase User, que se encarga de guardar los datos y generar los nombres de los ficheros y carpetas.

La clase User mandará los nombres generados a la clase EyeTrackerHandler, que utilizará más adelante.

4.2.2. Personalización de Configuración

Aquí, el usuario introducirá el texto a usar, la carpeta donde se guardarán los ficheros generados y el eye tracker para la prueba. Estos datos se enviarán a la clase EyeTrackerHandler.

Además, se le mandará el nombre del eye tracker a la clase DataManager, ya que, dependiendo de la decisión, se hará uso de un analizador de datos y otro.

4.2.3. Calibración y prueba

La calibración y la prueba se realizarán de la siguiente forma:

InitScreen llamará a la función test de EyeTrackerHandler para que comience la prueba.

Esta función generará un proceso paralelo al de la interfaz gráfica que se encargará del seguimiento ocular y la grabación.

4.2.4. Generación de resultados

La generación de resultados es dependiente a las clases EyeTrackerHandler, InitScreen y DataManager.

En la clase de InitScreen se realizará una llamada a la función de generación de gráficas de DataManager. Esta función recibirá como parámetros de entrada la dirección de ruta del fichero generado en la prueba y la dirección de ruta de la imagen utilizada en la prueba.

4.3. Interfaz de la aplicación

4.3.1. Interfaz final

Vamos a realizar una descripción y explicación de las distintas partes de la interfaz de la aplicación:

Inicio

Ventana principal de acceso al programa. El usuario puede acceder a cualquiera de las otras ventanas implementadas a través de los botones de la mitad inferior de la pantalla. En la parte superior hay un texto que explica el funcionamiento del eye tracking.

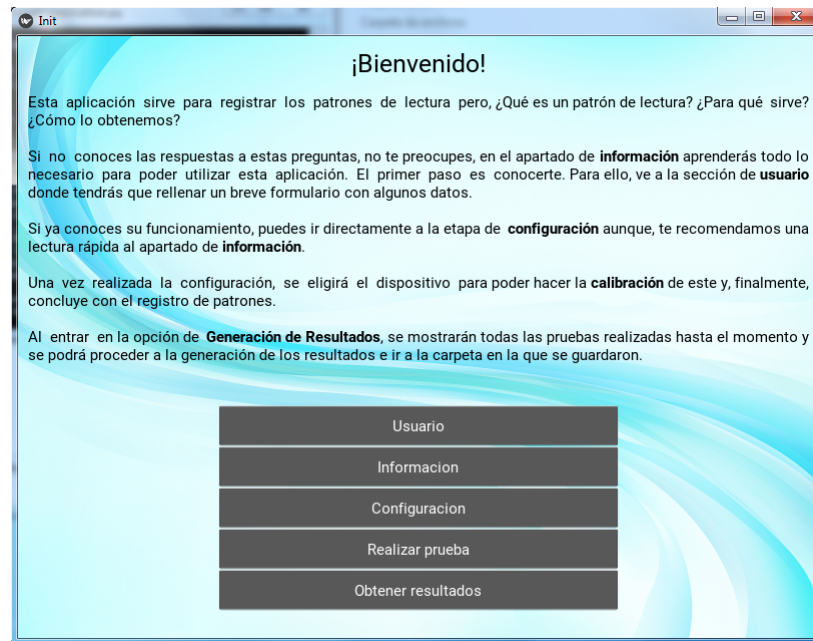


Figura 4.1: Ventana de inicio

Área de usuario

El usuario introduce sus datos personales para el desarrollo de la prueba (Nombre, apellidos y edad).

The image shows a software window titled 'Area de Usuario'. The background is light grey with a blue wave-like pattern at the bottom. The title 'Area de Usuario' is centered at the top. Below the title, there are three input fields. The first is labeled 'Nombre', the second 'Apellidos', and the third 'Edad'. Each label is to the left of its corresponding input field. At the bottom of the window, there is a dark grey button labeled 'Aceptar'.

Figura 4.2: Registro de usuario

Información

En esta ventana se puede visualizar información básica con respecto a la prueba y a cómo funciona el eye tracking de una forma muy básica con el objetivo de que el participante comprenda el funcionamiento de esta tecnología y en qué va a consistir la prueba que va a realizar.

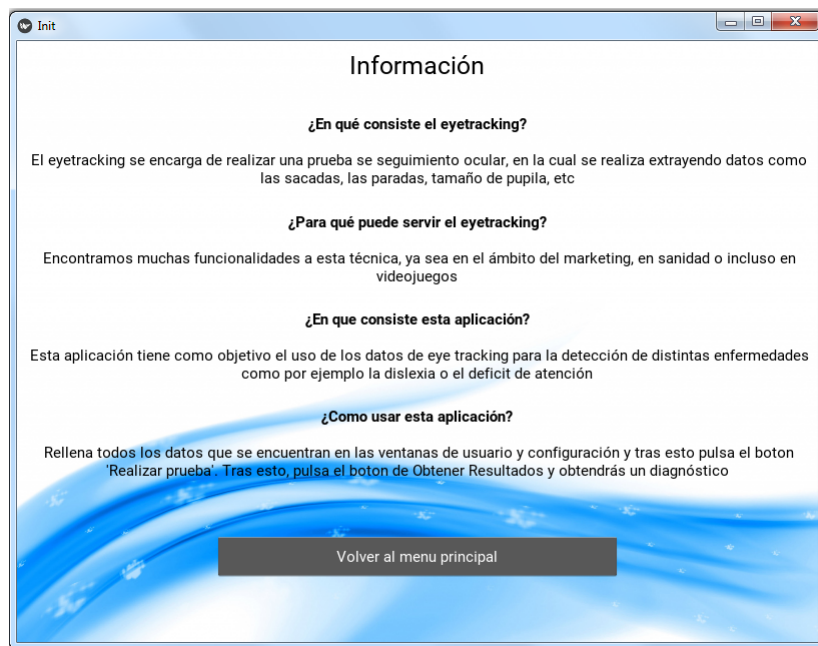


Figura 4.3: Ventana de información adicional

Configuración

La ventana de configuración recogerá los distintos datos de configuración de la prueba, micrófono a usar para grabar, el eye tracker a usar, por ahora se puede elegir entre Tobii y EyeTribe, el texto a visualizar, y la carpeta donde se quieren guardar los resultados.

Con respecto al texto que visualizar por parte del participante, tiene que cumplir los requisitos especificados en el diseño.



Figura 4.4: Ventana de configuración

Prueba

Cuando se pulsa el botón de la ventana de inicio **Realizar prueba**, un proceso paralelo al de la interfaz gráfica, se encarga de la calibración y la prueba a realizar con la imagen indicada en la ventana de configuración.

Como ya hemos comentado, la aplicación hace uso de las SDK para la calibración, es por ello que esta acción tiene distinta interfaz en los dos eye trackers, como vemos a continuación:

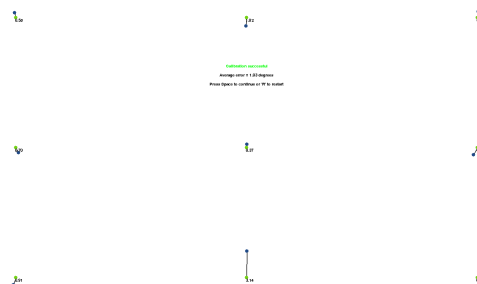


Figura 4.5: Calibración del eye tracker EyeTribe

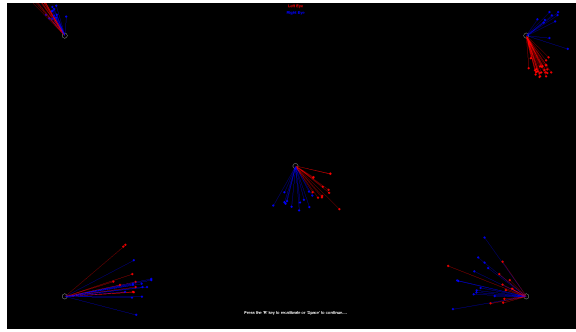


Figura 4.6: Calibración del eye tracker Tobii

Posteriormente, se mostrará el texto seleccionado y se realiza la prueba mientras se graba el audio.

Generación de resultados

Se generarán las gráficas a partir de los ficheros que contienen la información de la prueba.

Las gráficas mostrarán el tamaño de las pupilas y las coordenadas de toda la prueba, con el texto leído de fondo. Para que los datos se visualicen mejor, la aplicación omite las filas que han sido fallidas (Cuyas coordenadas son -1,-1).

5

Resultados

5.1. Objetivos de la evaluación

El objetivo de estas pruebas era la comprobación del buen funcionamiento de la aplicación con múltiples dispositivos de seguimiento ocular en distintos usuarios.

5.2. Pruebas

La prueba se ha realizado a 6 voluntarios, tres de ellos han realizado las pruebas con gafas, uno con lentillas y dos que no llevaban ni lentillas ni gafas. El voluntario que llevaba lentillas tuvo problemas en la calibración de ambos dispositivos, por lo que no se obtuvieron resultados válidos.

Se ha utilizado como texto a leer un extracto del cuento infantil Los tres cerditos, obtenido de la página web <http://www.cuentosinfantilesadormir.com/cuento-lostrescerditos.htm>.

Las pruebas consistirán en:

- Explicar al voluntario el objetivo del experimento.
- El voluntario realiza la calibración del primer dispositivo de seguimiento ocular con la aplicación implementada. Tras su calibración, el voluntario deberá de realizar una lectura del texto durante aproximadamente 15 segundos.

- Se utilizará la funcionalidad de la aplicación para generar las gráficas con los datos obtenidos, que serán mostrados por pantalla.
- Se repite la operación con el segundo dispositivo de seguimiento ocular.

5.3. Resultados

5.3.1. Voluntario 1

El primer voluntario llevaba gafas y tenía los ojos oscuros. No contó con ningún problema durante la calibración de ninguno de los dos dispositivos, obteniendo los resultados mostrados a continuación:

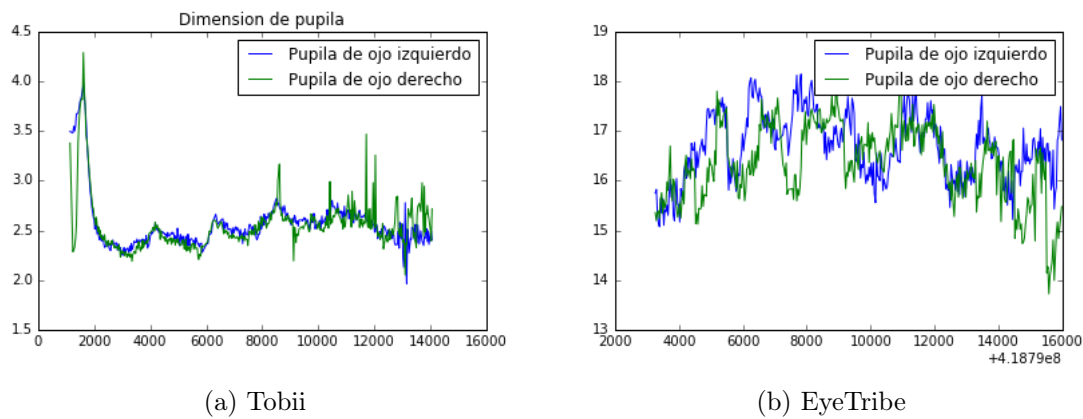


Figura 5.1: Tamaño de pupila

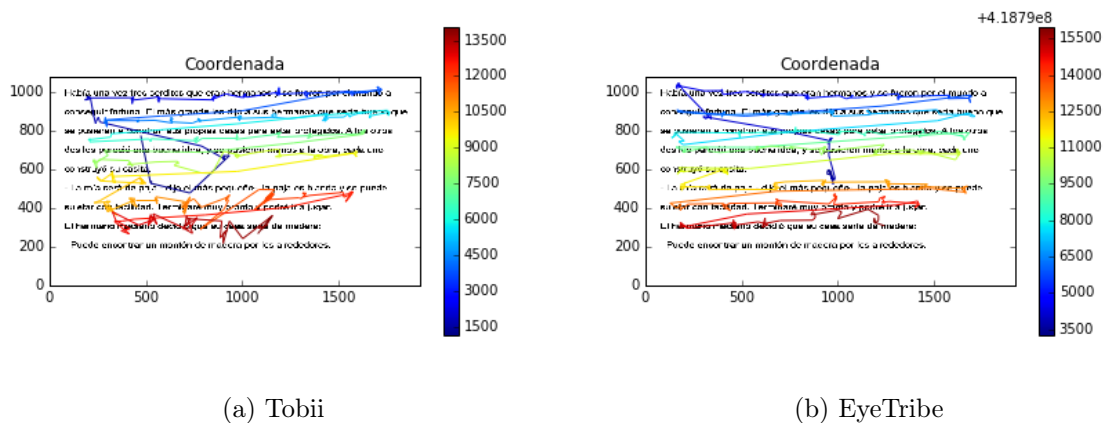


Figura 5.2: Coordenadas de ambos ojos

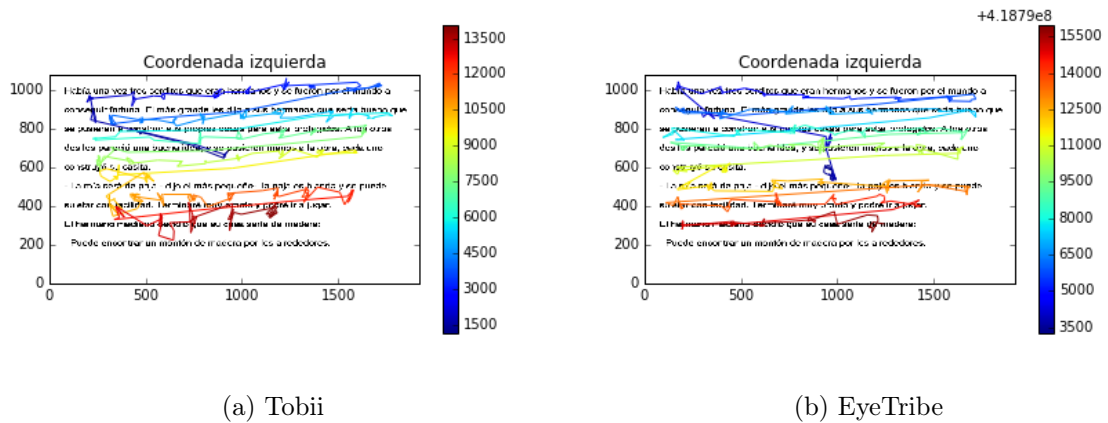


Figura 5.3: Coordenada de ojo izquierdo

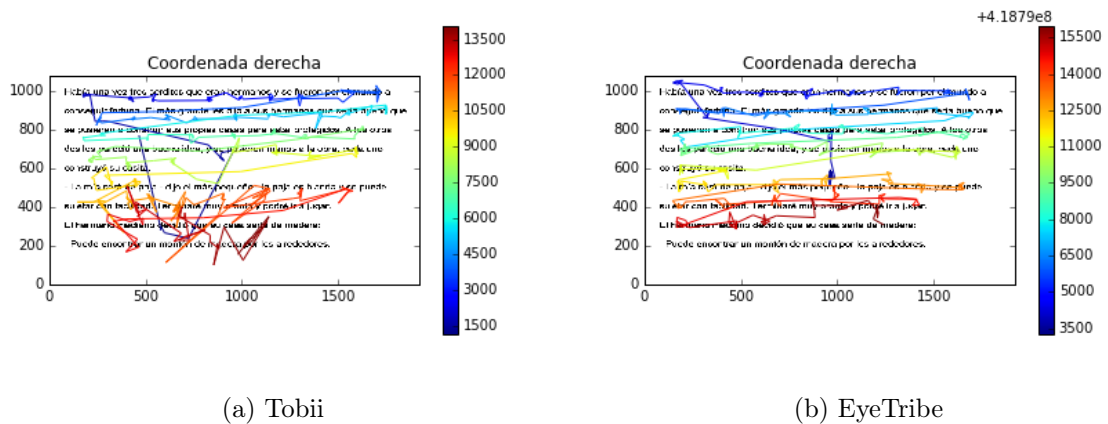
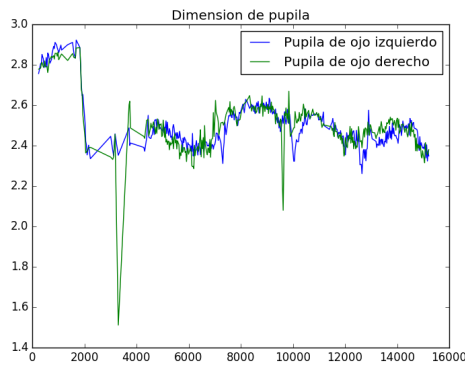


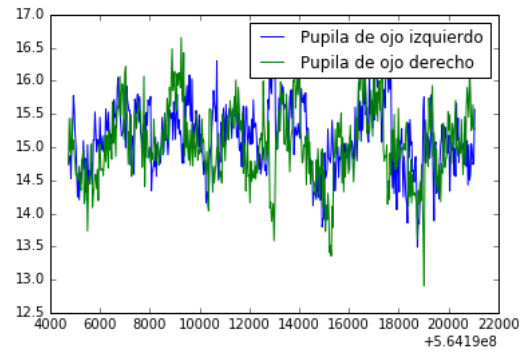
Figura 5.4: Coordenada de ojo derecho

5.3.2. Voluntario 2

El segundo voluntario no utilizó ni gafas ni lentillas durante la prueba y tenía los ojos claros. Este voluntario tampoco tuvo ningún problema durante la calibración y la realización de la prueba. Los resultados obtenidos fueron los siguientes:

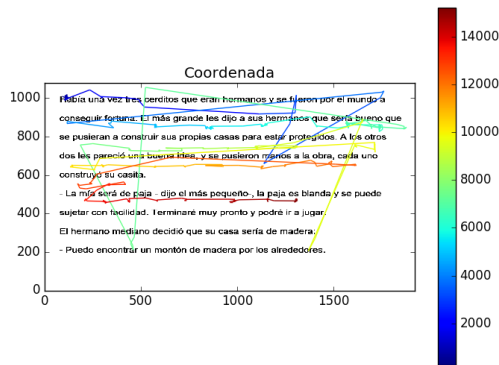


(a) Tobii

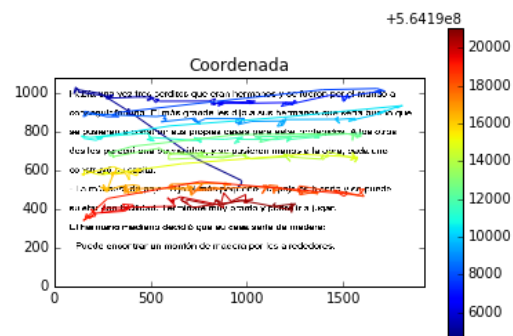


(b) EyeTribe

Figura 5.5: Tamaño de pupila

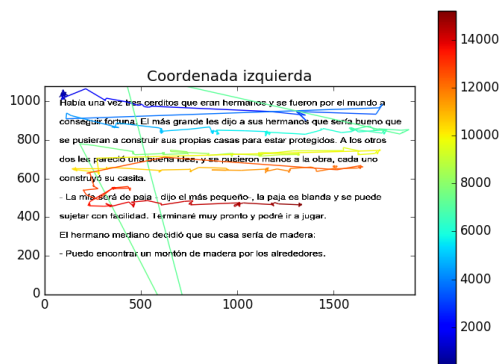


(a) Tobii

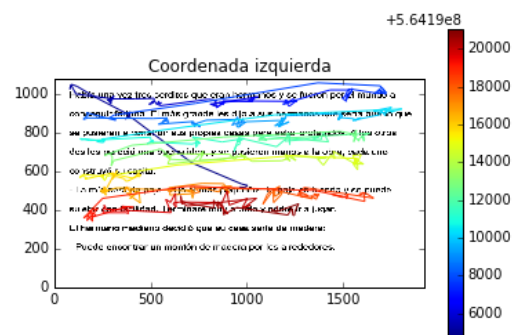


(b) EyeTribe

Figura 5.6: Coordenadas de ambos ojos

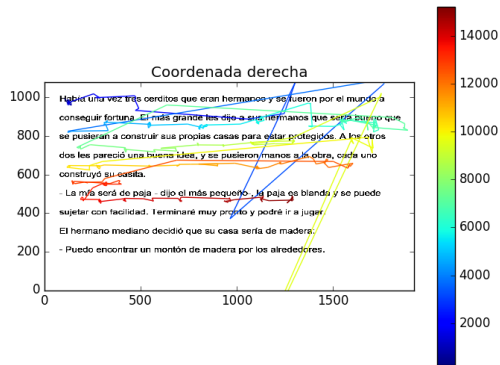


(a) Tobii

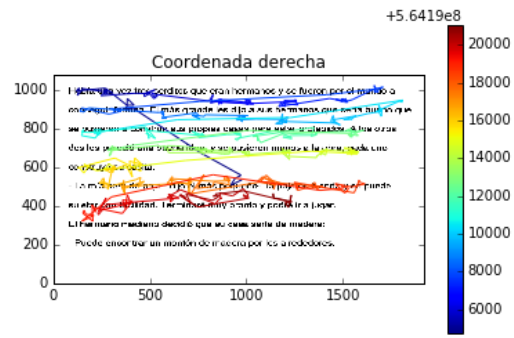


(b) EyeTribe

Figura 5.7: Coordenada de ojo izquierdo



(a) Tobii

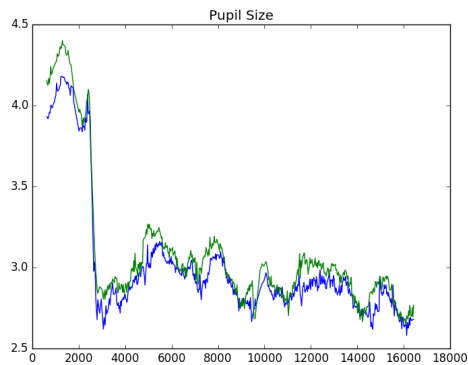


(b) EyeTribe

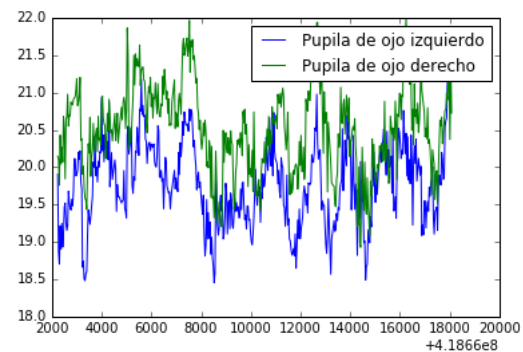
Figura 5.8: Coordenada de ojo derecho

5.3.3. Voluntario 3

El tercer voluntario, al igual que el primero, llevaba gafas y tenía ojos oscuros. Tampoco mostró ningún problema ni durante la calibración ni en la prueba.



(a) Tobii



(b) EyeTribe

Figura 5.9: Tamaño de pupila

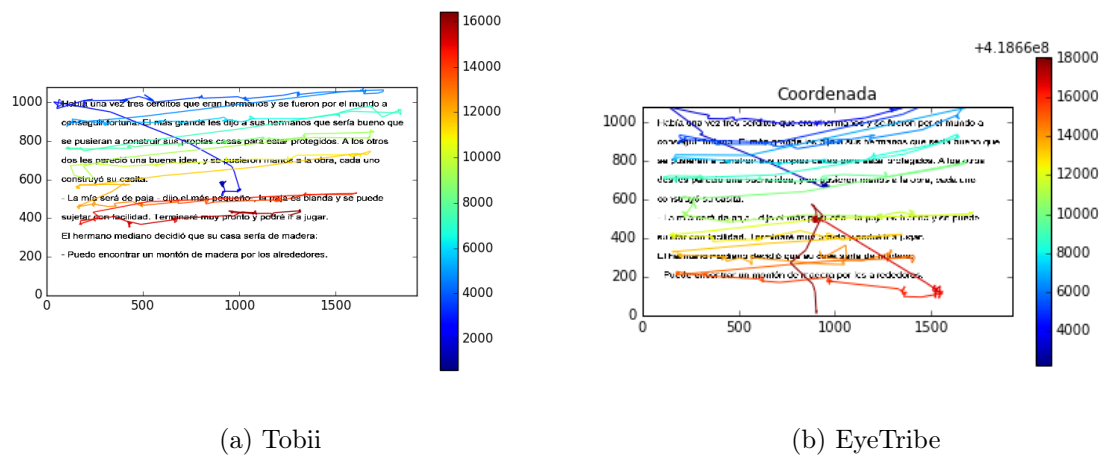


Figura 5.10: Coordenadas de ambos ojos

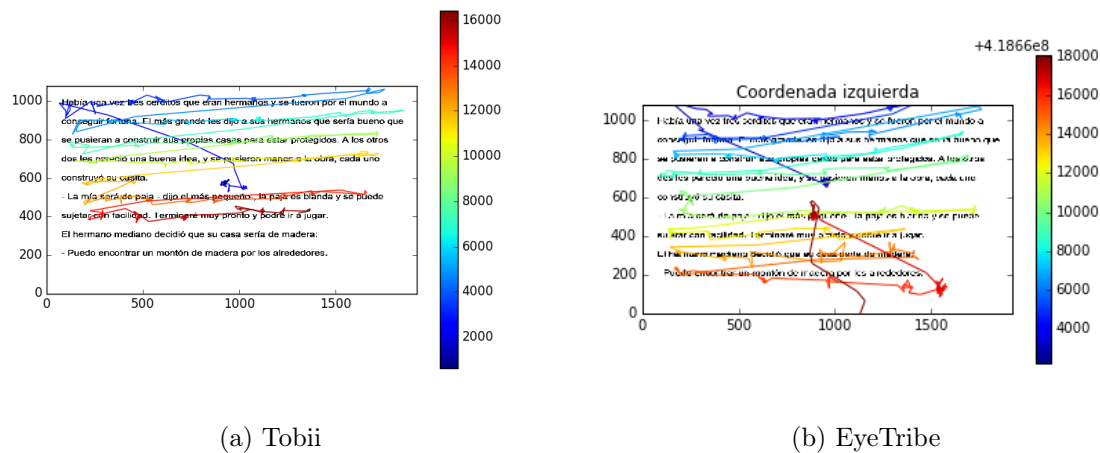


Figura 5.11: Coordenada de ojo izquierdo

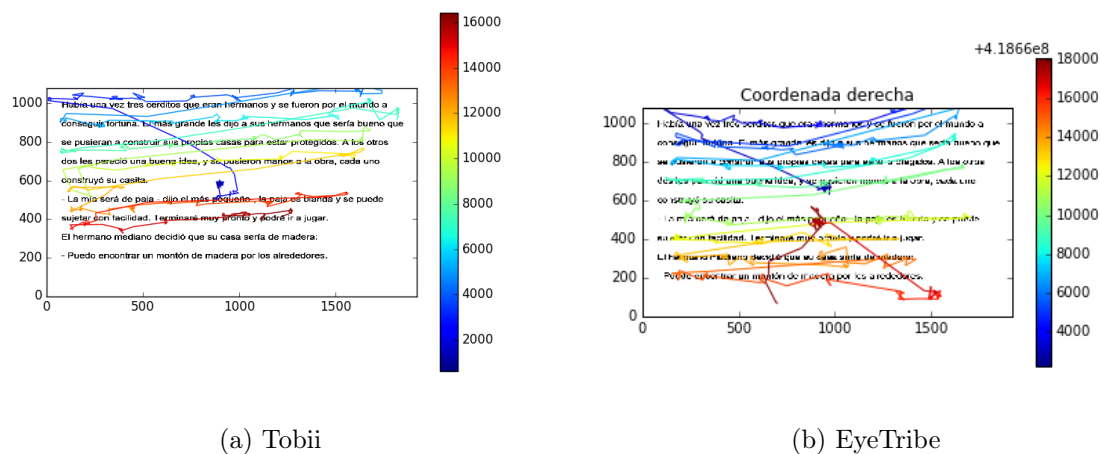


Figura 5.12: Coordenada de ojo derecho

5.3.4. Voluntario 4

El cuarto voluntario, de nuevo con gafas y ojos oscuros, tampoco mostro problemas durante la calibración. Sus resultados durante las pruebas fueron los siguientes:

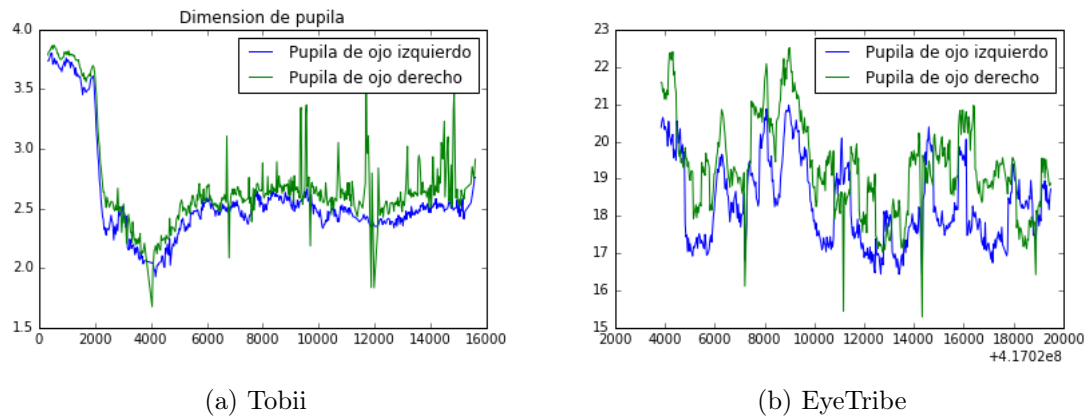


Figura 5.13: Tamaño de pupila

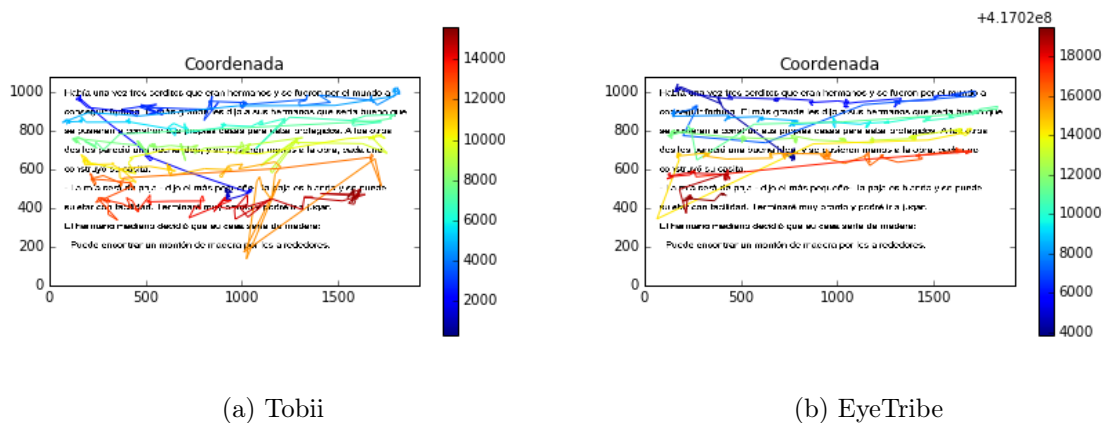


Figura 5.14: Coordenadas de ambos ojos

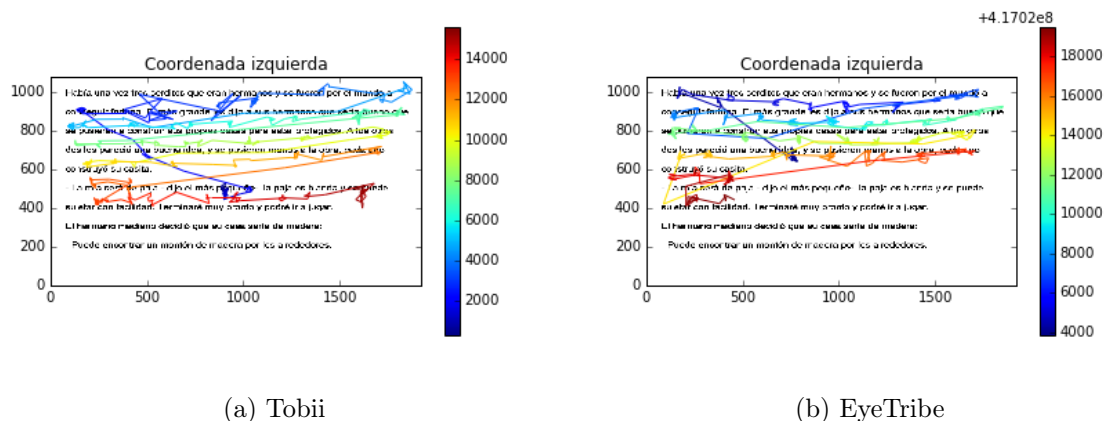


Figura 5.15: Coordenada de ojo izquierdo

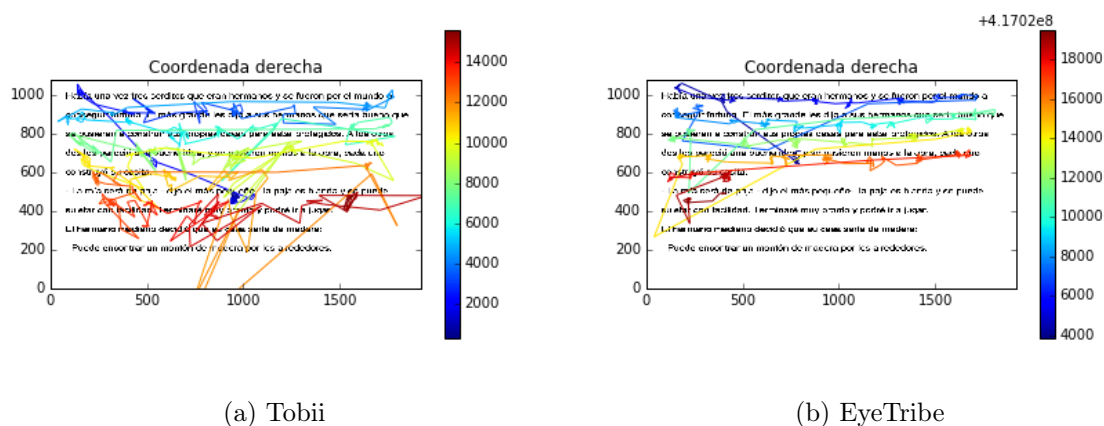
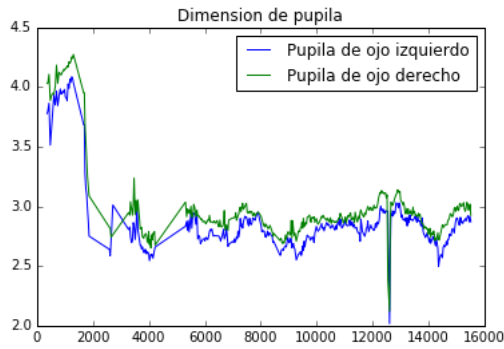


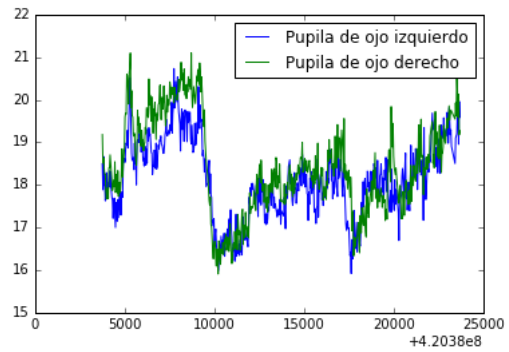
Figura 5.16: Coordenada de ojo derecho

5.3.5. Voluntario 5

Este voluntario realizó las pruebas sin gafas ni lentillas y tiene ojos oscuros. No tuvo problemas durante la calibración. Sus resultados fueron los siguientes:

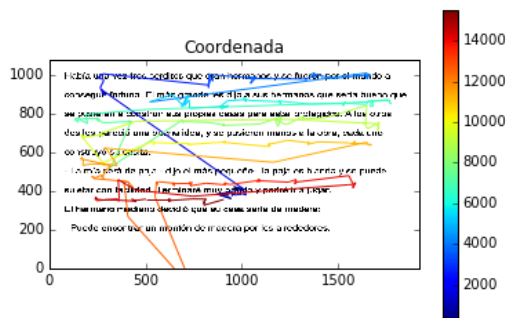


(a) Tobii

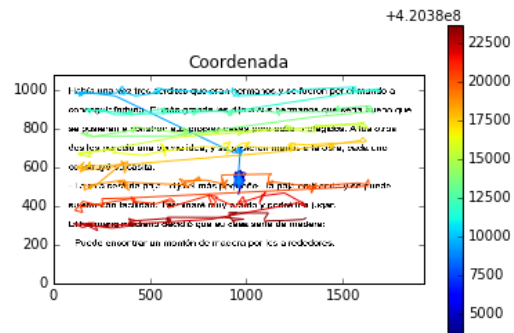


(b) EyeTribe

Figura 5.17: Tamanho de pupila

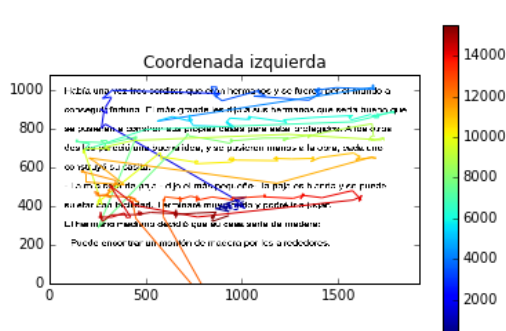


(a) Tobii

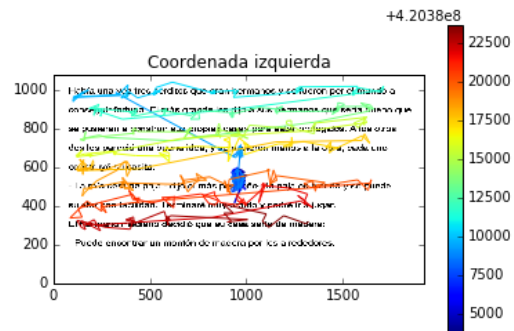


(b) EyeTribe

Figura 5.18: Coordenadas de ambos os olhos



(a) Tobii



(b) EyeTribe

Figura 5.19: Coordenada de olho esquerdo

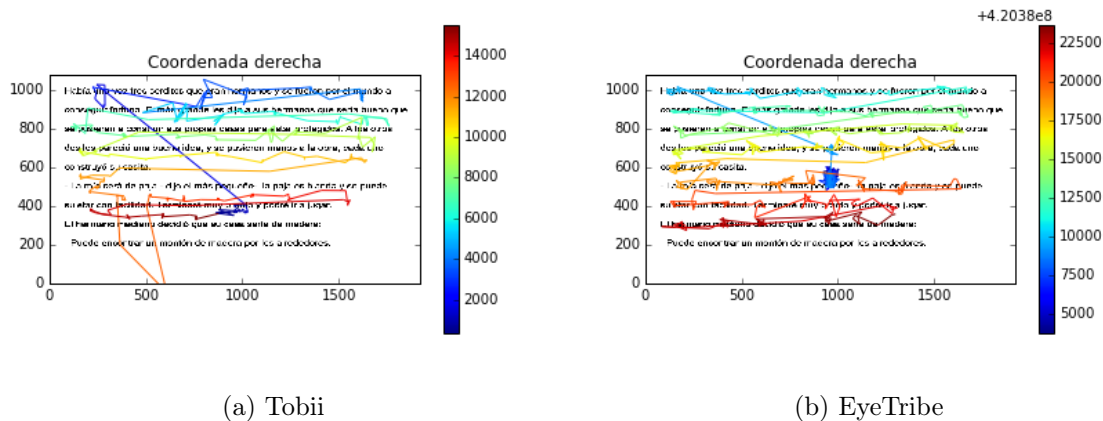


Figura 5.20: Coordenada de ojo derecho

5.4. Análisis de los resultados

Las gráficas generadas por nuestro programa muestran las coordenadas a las que ha mirado el voluntario durante la prueba, con el texto leído mostrado de fondo. La variación de color indica la variación temporal entre coordenadas.

Las gráficas de tamaño de pupila mostrarán el tamaño de la pupila de los dos ojos por separado, la unidad de tiempo de las graficas será milisegundos.

Para el análisis solo hemos tenido en cuenta los resultados de los cinco primeros voluntarios, al no haber podido aportar datos coherentes el sexto voluntario.

La generación de estas gráficas llevada a cabo por nuestra aplicación es un indicador de la capacidad de trabajo con múltiples dispositivos de seguimiento ocular.

Podemos observar que, dentro de todas las comparaciones entre las gráficas, la más destacada es la del tamaño de pupila. Esto es debido a lo explicado anteriormente con respecto a las unidades utilizadas para medir este campo por cada eye tracker. Aun con las claras diferencias, observamos ciertas similitudes como puede ser que el ojo de mayor tamaño coincide en ambos ejemplos y que se producen unas oscilaciones bastante parecidas en ambos.

Por otro lado, con respecto a las coordenadas. Observamos que aunque son diferentes, aproximadamente siguen el mismo comportamiento, siendo más detalladas con el eye tracker de Tobii, al ser este más preciso que el eye tracker de EyeTribe. Esto podría ser indicador de detección patrón de lectura del individuo.

6

Conclusiones

6.1. Conclusiones

El objetivo de este Trabajo de Fin de Grado era la realización de una aplicación capaz de trabajar con varios dispositivos de seguimiento ocular y así facilitar la detección de trastornos que tengan una clara manifestación en el registro de los movimientos oculares.

Podemos observar según las pruebas de los resultados y lo indicado en el capítulo referente al desarrollo de la aplicación que hemos cubierto satisfactoriamente el objetivo de que la aplicación funcione con varios dispositivos de distintas marcas y modelos.

Además, como hemos indicado en el análisis de los resultados, vemos como con los distintos voluntarios de las pruebas tienen patrones de lectura específicos, aunque siguen una trayectoria muy similar, no siendo muy diferentes los datos obtenidos entre los dos eye trackers. Es por ello que es esperable que la aplicación desarrollada sea capaz de utilizarse para registrar las diferencias significativas entre una persona sana y una persona que padezca una enfermedad con una manifestación en los movimientos oculares.

6.2. Trabajo futuro

La trayectoria futura a la que puede dirigirse esta aplicación podría ser la incorporación de herramientas de aprendizaje automático optimizadas para realizar la clasificación de las series temporales registradas [18, 19]. Por su diseño, la herramienta puede incorporar fácilmente nuevos dispositivos de registro, sobre todo atendiendo a los de más bajo coste que pueden favorecer la diseminación de esta tecnología para facilitar la caracterización

de los patrones de movimiento ocular y la **recomendación de diagnóstico** de múltiples enfermedades. Idealmente, este tipo de aplicaciones podrían ejecutarse en los dispositivos de lectura de cualquier colegio o centro educativo para implementar sistemas de detección temprana de patologías.

Bibliografía

- [1] D Rozado y col. «Controlling a Smartphone Using Gaze Gestures as the Input Mechanism». En: *Human-Computer Interaction* 30.1 (2015), págs. 34-63. ISSN: 0737-0024. DOI: 10.1080/07370024.2013.870385. URL: <http://dx.doi.org/10.1080/07370024.2013.870385>.
- [2] Andrew T. Duchowski. *Eye Tracking Methodology*. Cham: Springer International Publishing, 2017. ISBN: 978-3-319-57881-1. DOI: 10.1007/978-3-319-57883-5. URL: <http://link.springer.com/10.1007/978-3-319-57883-5>.
- [3] Hajra Ashraf y col. «Eye-tracking technology in medical education: A systematic review». En: *Medical teacher* 40.1 (2018), págs. 62-69.
- [4] Keith Rayner. «Eye movements in reading and information processing: 20 years of research.» En: *Psychological Bulletin* 124.3 (1998), págs. 372-422. ISSN: 1939-1455. DOI: 10.1037/0033-2909.124.3.372. URL: <http://doi.apa.org/getdoi.cfm?doi=10.1037/0033-2909.124.3.372>.
- [5] Christina Spiesel. «Reflections on reading: Words and pictures and law». En: *International Journal of Law in Context* 2.3 (2006), págs. 305-320. ISSN: 17445531. DOI: 10.1017/S1744552306003077.
- [6] Andrew T Duchowski. «A breadth-first survey of eye-tracking applications». En: *Behavior research methods, instruments, and computers* 34.4 (2002), págs. 455-470. ISSN: 0743-3808. DOI: 10.3758/BF03195475.
- [7] Kyle Krafka y col. «Eye Tracking for Everyone». En: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, págs. 2176-2184. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.239. URL: <http://ieeexplore.ieee.org/document/7780608/>.
- [8] Zohreh Sharafi, Zéphyrin Soh y Yann-Gaël Guéhéneuc. «A Systematic Literature Review on the Usage of Eye-tracking in Software Engineering». En: (). URL: <http://www.ptidej.net/publications/documents/IST15.doc.pdf>.
- [9] Areej Al-Wabil, Panayiotis Zaphiris y Stephanie Wilson. «Examining visual attention of dyslexics on web navigation structures with eye tracking». En: *2008 International Conference on Innovations in Information Technology, IIT 2008* (2008), págs. 717-721. DOI: 10.1109/INNOVATIONS.2008.4781673.

- [10] Keith Rayner y Charles Clifton. «Language processing in reading and speech perception is fast and incremental: Implications for event-related potential research». En: *Biological Psychology* (2009). ISSN: 03010511. DOI: 10.1016/j.biopsycho.2008.05.002.
- [11] Stéphanie Bellocchi y col. «I can read it in your eyes: What eye movements tell us about visuo-attentional processes in developmental dyslexia». En: *Research in Developmental Disabilities* 34.1 (2013), págs. 452-460. ISSN: 0891-4222. DOI: 10.1016/J.RIDD.2012.09.002. URL: <https://www.sciencedirect.com/science/article/pii/S089142221200234X>.
- [12] Luz Rello y Ricardo Baeza-Yates. «Good fonts for dyslexia». En: *ASSETS 2013: Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility* (2013), págs. 1-8. DOI: 10.1145/2513383.2513447.
- [13] Amy S Desroches, Marc F Joanisse y Erin K Robertson. «Specific phonological impairments in dyslexia revealed by eyetracking». En: (). DOI: 10.1016/j.cognition.2005.09.001.
- [14] Richard Kunert y Christoph Scheepers. «Speed and accuracy of dyslexic versus typical word recognition: An eye-movement investigation». En: *Frontiers in Psychology* 5.SEP (2014), págs. 1-16. ISSN: 16641078. DOI: 10.3389/fpsyg.2014.01129.
- [15] Andrew T. Duchowski. «A breadth-first survey of eye-tracking applications». En: *Behavior Research Methods, Instruments, & Computers* 34.4 (2002), págs. 455-470. ISSN: 0743-3808. DOI: 10.3758/BF03195475. URL: <http://www.springerlink.com/index/10.3758/BF03195475>.
- [16] Jessica Beltrán y col. «Computational Techniques for Eye Movements Analysis towards Supporting Early Diagnosis of Alzheimer's Disease: A Review». En: *Computational and Mathematical Methods in Medicine* 2018 (2018), págs. 1-13. ISSN: 1748-670X. DOI: 10.1155/2018/2676409. URL: <https://www.hindawi.com/journals/cmmm/2018/2676409/>.
- [17] Edwin S. Dalmaijer, Sebastiaan Mathôt y Stefan Van der Stigchel. «PyGaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments». En: *Behavior Research Methods* 46.4 (2014), págs. 913-921. ISSN: 1554-3528. DOI: 10.3758/s13428-013-0422-2. URL: <http://link.springer.com/10.3758/s13428-013-0422-2>.
- [18] David Rozado, Francisco B. Rodriguez y Pablo Varona. «Optimizing Hierarchical Temporal Memory for Multivariable Time Series». En: *Artificial Neural Networks – ICANN 2010*. Ed. por Konstantinos Diamantaras, Wlodek Duch y Lazaros S. Iliadis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, págs. 506-518. ISBN: 978-3-642-15822-3. DOI: https://doi.org/10.1007/978-3-642-15822-3_62.
- [19] David Rozado, Francisco B. Rodriguez y Pablo Varona. «Extending the bioinspired hierarchical temporal memory paradigm for sign language recognition». En: *Neurocomputing* 79 (2012), págs. 75 -86. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2011.10.005>. URL: <http://www.sciencedirect.com/science/article/pii/S0925231211006230>.

Apéndices



Extensión de diseño

En este apéndice mostraremos los requisitos del programa y los diagramas realizados para una mejor visualización del diseño.

A.1. Requisitos

A.1.1. Requisitos funcionales

Inserción de datos. La aplicación debe permitir al usuario la inserción de distintos datos e información referente a la prueba a realizar.

Calibración de los dispositivos de seguimiento ocular. La aplicación debe de permitir la calibración del dispositivo de seguimiento ocular antes de realizar la prueba.

Generación de una prueba de seguimiento ocular. La aplicación debe generar una prueba de seguimiento ocular que el usuario pueda utilizar.

Grabación de audio. La aplicación tiene que generar un fichero de audio correspondiente al momento de realización de la prueba.

Representación de gráficas de datos y resultados. La aplicación debe de ser capaz de generar gráficas con los datos obtenidos a partir de la prueba de seguimiento ocular realizada con antelación.

A.1.2. Requisitos no funcionales

Interfaz üserfriendly. La aplicación debe de tener una interfaz gráfica amigable para el usuario y permita un uso fluido de la misma.

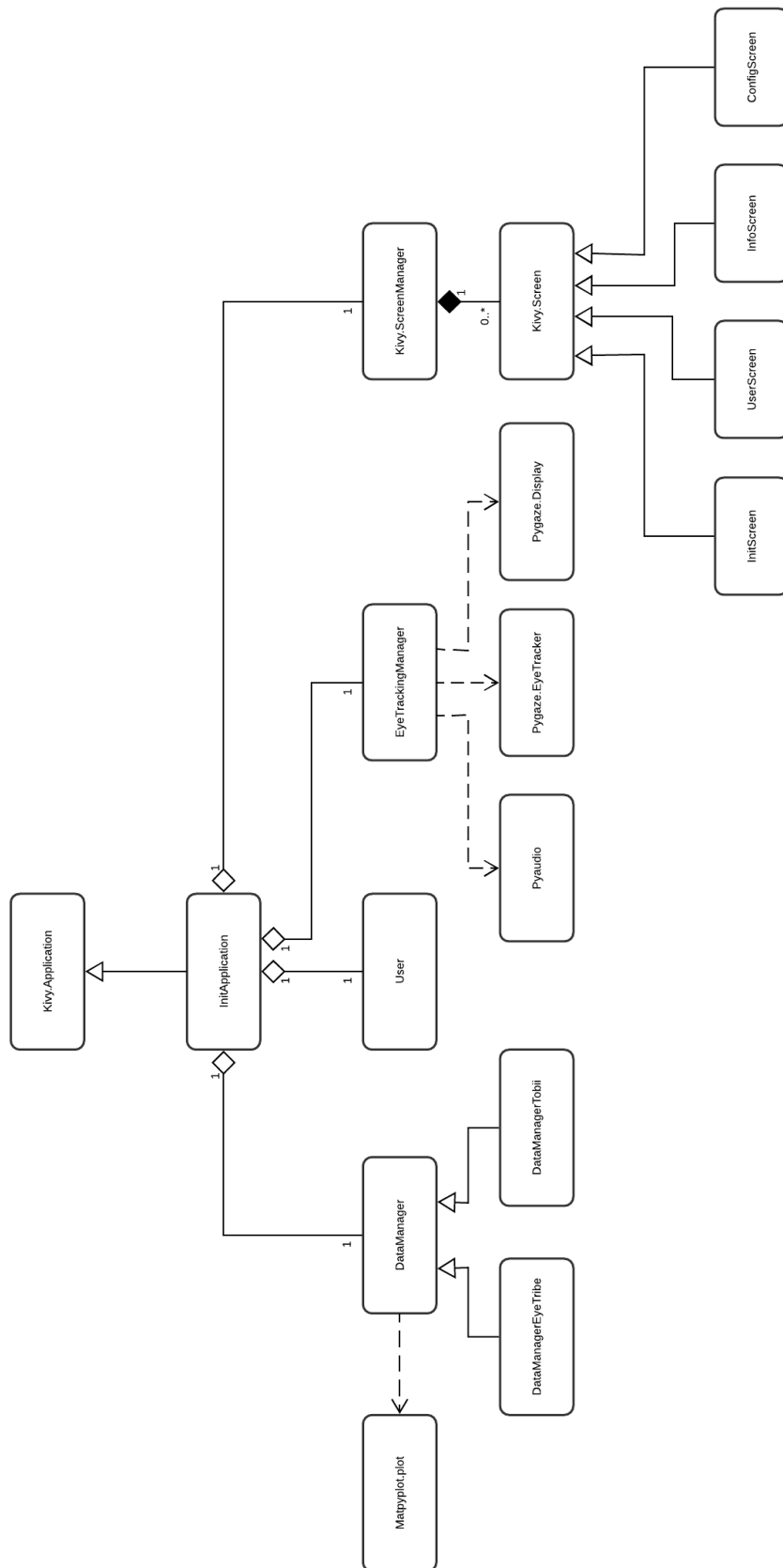
Modularidad. Realizaremos una arquitectura modular, siendo muy importante en la clase encargada de la manipulación de datos y la generación de gráficas, al ser la parte que más puede variar por los distintos ficheros de los distintos dispositivos.

A.2. Diagrama de clases

A continuación se mostrará el diagrama de clases de la aplicación:

A.3. Código obtenido

Facilitamos el código desarrollado durante este Trabajo de Fin de Grado



B

Guia de Instalación

En este apartado explicaremos qué instalar para poder hacer uso de nuestra aplicación. Tenemos que dividir en varias partes el proyecto, en primer lugar la interfaz gráfica, la parte de eye tracking, la de grabación y finalmente la de generación de gráficas.

Para trabajar sobre este proyecto, como ya se ha comentado en el documento, hemos hecho uso de la herramienta WinPython, de la cual podemos obtener más información en su página web <http://winpython.sourceforge.net/>. Hemos hecho uso de ella para facilitar la obtención de un entorno de trabajo para el desarrollo de la aplicación.

B.1. Interfaz gráfica

Como hemos comentado anteriormente en el cuerpo del documento, para la implementación de la interfaz gráfica hemos hecho uso del framework Kivy.

Este fácil y manejable framework cuenta con una instalación en Windows también muy sencilla, indicada en la web <https://kivy.org/docs/installation/installation-windows.html#install-win-dist>.

B.2. Eyetracking

PyGaze cuenta con múltiples dependencias, pudiéndose prescindir de alguna de ellas dependiendo del uso que se dé de la aplicación.

Para explicar esto de forma correcta expondremos un poco el funcionamiento de la

librería.

Esta librería cuenta con varias clases y ficheros que se utilizan para la interacción entre el ordenador y el dispositivo de seguimiento ocular, entre todas estas clases y ficheros encontramos una que se encarga de la configuración que se utilizará durante el manejo del dispositivo, este fichero es `defaults.py`, el cual podemos observar detenidamente en la web <https://github.com/esdalmaijer/PyGaze/blob/master/pygaze/defaults.py>.

Como podemos observar, este fichero se encarga de la configuración de campos tanto referentes al eye tracker a usar como a nivel del fichero de log, el tamaño de la pantalla o la generación de la interfaz gráfica. Uno de los campos que encontramos en esta lista es "DISPTYPE", en el cual se puede indicar que se va a usar Pygame o PsychoPy. Otro campo que nos interesa de esta lista sería el denominado "TRACKERTYPE", este campo se encarga de indicar que eye tracker se está utilizando durante la prueba. Podemos observar que hay múltiples opciones, tantas como marcas compatibles de eye trackers puede utilizar la librería.

Sabiendo esto ya podemos continuar con la explicación.

Las dependencias las podemos encontrar en esta web <http://www.pygaze.org/installation/>.

Sabiendo ahora lo explicado anteriormente y conociendo las dependencias a instalar, de esas dependencias prescindimos de las siguientes:

- PsychoPy. En lo que respecta a nuestra aplicación, para la interacción entre el eye tracker y la misma, hacemos uso de la librería PyGame, la cual nos facilita las acciones de calibración y la inserción de texto en el tiempo de prueba.
- La dependencia de Tobii especificada en la web de instalación. Esta dependencia se refería a la anterior SDK del fabricante, ahora, haciendo uso de la nueva SDK podemos prescindir de esa instalación.
- Todo las dependencias referentes a eye trackers que no sean EyeTribe o Tobii. Actualmente trabajamos con los dos indicados, por ello, las otras dependencias podrían llegar a ser opcionales en caso de hacer uso solo de estos dos eye trackers.

B.3. Audio

Con respecto a la parte de audio, la librería PyAudio ha sido nuestra herramienta para desarrollar la implementación de este módulo en nuestra aplicación. Gracias al uso de WinPython no nos hemos tenido que preocupar de su instalación.

B.4. Generación de gráficas

Al igual que con PyAudio, la librería matplotlib usada ya estaba instalada.

Tras la instalación de todos estos componentes obtenemos un entorno que nos permite el desarrollo y uso de nuestra aplicación.

Para facilitar el trabajo de generar este entorno, facilitamos el que hemos usado durante la implementación <https://mega.nz/#!HkBkBSID!0aH1uATadhnIifFbmkHRM6bw7aCXnUCz6FWZqHu-nw>



Información extendida de los eye trackers usados

C.1. Comparación de eye trackers

Los dos eye trackers usados cuentan con diferencias en distintas categorías y campos. Para facilitar su visualización mostramos estas diferencias en la tabla de a continuación.

-	EyeTribe	Tobii
Muestreo	Muestreo de 30 y 60 Hz	30 Hz
Precisión	0.5°	0.4°
Latencia	<20 ms en 60 Hz	48–67 ms
Calibración	5, 9 y 12 puntos	5 puntos
Rango de funcionamiento	45–75 cm	50–100 cm
Tamaño de pantalla	A partir de 24"	A partir de 27"
SDK	C++, C# y Java	.NET, Python, Matlab/Octave y C
Precio	99\$	20,000€

Tabla C.1: Tabla de comparación de los dispositivos utilizados durante la prueba

C.2. SDK

Vamos a explicar la instalación realizada para poder usar las SDK de los distintos dispositivos

C.2.1. EyeTribe

Como se ha indicado en la memoria, tenemos que acceder a la web <https://github.com/EyeTribe/sdk-installers/releases> para descargarnos la SDK, la cual constara de un fichero ejecutable a descargar. Tras realizar la instalación, tendremos dos nuevos programas en nuestro ordenador, EyeTribe Server y EyeTribe UI. Para poder utilizar el eye tracker EyeTribe tenemos que tener abierto el programa EyeTribe Server durante su uso además de nuestra aplicación.

C.2.2. Tobii

Para utilizar la SDK de Tobii de nuevo tenemos que acceder a la página de Tobii <https://www.tobiipro.com/product-listing/tobii-pro-sdk/#Download>. Seleccionamos la versión para Python. Tras rellenar un formulario donde introducimos una serie de datos personales se nos descargará un fichero comprimido donde encontraremos las versiones de 32 y 64 bits. El contenido de la carpeta escogida la copiamos en la carpeta de `site_packages` de Python.